Instrument-specific harmonic atoms for mid-level music representation

Pierre Leveau, Emmanuel Vincent, Gael Richard, Laurent Daudet

Instrument-Specific Harmonic Atoms for Mid-Level Music Representation

Pierre Leveau, Emmanuel Vincent, Gaël Richard, *Senior Member, IEEE,* and Laurent Daudet, *Member, IEEE*

Abstract

Several studies have pointed out the need for accurate mid-level representations of music signals for information retrieval and signal processing purposes. In this article, we propose a new mid-level representation based on the decomposition of a signal into a small number of sound atoms or molecules bearing explicit musical instrument labels. Each atom is a sum of windowed harmonic sinusoidal partials whose relative amplitudes are specific to one instrument, and each molecule consists of several atoms from the same instrument spanning successive time windows. We design efficient algorithms to extract the most prominent atoms or molecules and investigate several applications of this representation, including polyphonic instrument recognition and music visualization.

Index Terms

Mid-level representation, sparse decomposition, music information retrieval, music visualization.

I. Introduction

When listening to music, humans experience the sound they perceive in view of their prior knowledge, using a collection of global properties, such as musical genre, tempo and orchestration, as well as

Manuscript received February 14, 2007. This work has been partially supported by the European Commission under contract FP6-027026-K-SPACE, CNRS and EPSRC grant GR/S75802/01.

Pierre Leveau and Laurent Daudet are with the University Pierre et Marie Curie-Paris 6, Institut Jean Le Rond d'Alembert, LAM team, 11 rue de Lourmel, F-75015 Paris, France (e-mail: leveau@lam.jussieu.fr, daudet@lam.jussieu.fr).

Emmanuel Vincent is with the METISS Project, IRISA-INRIA, Campus de Beaulieu, 35042 Rennes cedex, France (e-mail: emmanuel.vincent@irisa.fr).

Pierre Leveau and Gaël Richard are with the Département TSI, GET-ENST, 37-39 rue Dareau, F-75014 Paris, France (email: gael.richard@enst.fr).

more specific properties, such as the timbre of a particular instrument. Bridging the gap between audio waveforms and such high-level properties constitutes the aim of *semantic audio analysis*, which has attracted a lot of research effort recently. Ultimately, machine listening systems with close-to-human performance would lead to improvements for many signal processing applications, including user-friendly browsing of music archives and interactive sound modification.

On the one hand, starting from the audio waveform, a large number of *low-level features* have been proposed for the description of timbre and harmony within short time frames, such as the popular Mel-Frequency Cepstral Coefficients (MFCC) [1], chroma vectors [2] and other features standardized in MPEG-7 [3]. Based on these features, algorithms have been developed for genre or artist classification [4], instrument recognition [5], key finding [6] and structural segmentation [7]. Recent algorithms achieve good success rates, but seem to have reached a performance ceiling such that increasing complexity no longer significantly improves performance. This experimental observation can be partly explained by two factors [4]. Firstly, low-level features only provide a rough description of *polyphonic* (*i.e.* multi-instrumental) data since they model the input sound as a whole, whereas humans are generally able to describe, to some extent, each instrument separately. Secondly, these features, being defined on short time frames, do not easily account for long-term dependencies or rare events. Existing algorithms typically use "bag-of-frames" approaches: features are extracted at fixed time lags, each lag corresponding to a frame, sometimes with additional derivative or variance features. Hence, a given musical extract is described by a collection of framewise features called a "bag of frames". Then, classes (e.g. instruments, groups of instruments, musical genres) are modelled in this feature space using machine learning algorithms such as K-Nearest Neighbors [8], Gaussian Mixture Models [8], [9], or Support Vector Machines [9]. By contrast, humans may assess temporal variations at different time scales for each instrument and discriminate similar data based on time-localized cues observed in a few time frames only.

On the other hand, a significant amount of work has been devoted to the processing of music in a symbolic framework, most commonly using the Musical Instrument Digital Interface (MIDI) as the input format [10], [11], [12]. This score-like format exhibits several advantages over audio, since it is based on a considerably reduced amount of data, while incorporating much higher-level information in the form of note events and orchestration. This allows the use of advanced musicological models that may improve performance for certain tasks [10]. However, the main limitation of MIDI is that it loses some fine information available in audio signals such as frequency and amplitude modulations and spectral envelopes, which may be valuable for other tasks.

Ideally, we would like to enjoy the best of both worlds by jointly processing audio and symbolic repre-

sentations. However, most music is available in audio format only and perfect polyphonic audio-to-MIDI converters are out of reach of today's technology [13]. An alternative solution is to derive intermediate signal representations emphasizing some semantic properties of music without seeking to estimate actual musical scores. To address the limitations of low-level features, these *mid-level representations* should fulfill two goals:

- describing instruments separately as much as possible,
- incorporating long-term structures.

This idea was introduced in [14] along with a possible mid-level representation involving different parametric sound objects, including "weft" objects consisting of harmonic sinusoidal partials. Other mid-level representations were proposed more recently for rhythmic [15] and harmonic content [16]. A limitation of these various representations is that they do not provide *orchestration* information, *i.e.* the instruments that are playing. This information is however crucial for genre classification [10] and would also allow separate visualization or processing of each instrument. An interesting approach that includes knowledge on instruments to represent signals has been made in [17]. The author introduced a non-resynthesizable representation that shows instrument presence probabilities as a function of time and pitch range, without onset detection or pitch estimation.

In this article, we propose a new mid-level representation of music signals that incorporates explicit instrument labels and intends to provide a *single* front-end for many information retrieval and signal processing tasks. This representation is derived from recent advances in the field of sparse approximation concerning the modeling of signal structures. The signal is decomposed into a small number of sound *atoms* or *molecules*, where each atom is a sum of windowed harmonic sinusoidal partials and each molecule is a group of atoms spanning successive time windows. This signal model aims at representing harmonic instruments, namely wind instruments, bowed strings instruments or tonal parts of singing voice. The additivity of the signal model makes it directly applicable to chords and multi-instrument pieces. As such, it is not suited for non-harmonic instruments (e.g. drums) and slightly inharmonic instruments (e.g. piano). However, by taking advantage of the flexibility of sparse representations, it would be possible to include other types of atoms designed for these specific sources. In this study, each atom is labelled with a specific instrument by prior learning of the amplitudes of its partials on isolated notes. The instantaneous amplitudes and frequencies of the partials and their temporal variations can provide additional timbre information.

Our goal is to get representations that exhibit some information on the played notes, such as intensity,

pitch, onset, offset and timbre. Clearly, more complex musicological models would be needed for accurate score transcription, where the goal is to minimize the estimation errors of the aforementioned parameters [13]. Nevertheless the representations described in this article still allow the inference of higher-level knowledge, such as the orchestration, the pitch range, the most typical intervals between notes, etc.

The article is organized as follows. In Section II, we present the rationale for the signal model and provide a mathematical definition of atoms and molecules. We subsequently design efficient algorithms to extract the most prominent atoms or molecules (Sections III and IV) and to learn the model parameters (Section V). In Section VI, we illustrate two applications of the proposed representation: music visualization and music instrument recognition on solo and duo pieces. Finally, we conclude in Section VII and provide perspectives on further research.

II. Signal model

Generally speaking, the goal of sparse approximation is to represent a discrete-time signal $x(t)$ as a weighted sum of atoms $h_\lambda(t)$, taken from a fixed *dictionary* $\mathcal{D} = \{h_\lambda(t)\}_\lambda$, plus a residual $r(t)$

$$x(t) = \sum_{\lambda \in \Lambda} \alpha_\lambda h_\lambda(t) + r(t). \tag{1}$$

where Λ is a finite set of indexes λ. The precision of the approximation can be measured by the Signal-to-Residual Ratio (SRR) in decibels (dB) defined by $\mathrm{SRR} = 10 \log_{10}(\sum_t x(t)^2 / \sum_t r(t)^2)$. The term *sparse* refers to the desirable goal for a decomposition that the number $\mathrm{card}(\Lambda)$ of selected atoms be as low as possible for a given SRR and much lower than the length of the signal in number of samples. When the atoms are similar to the signal, high sparsity and high SRR can be achieved at the same time.

Sparse decomposition provides a natural framework for mid-level representation. Indeed, the set of atoms representing the observed signal may be partitioned into multiple subsets, where each subset represents a different instrument and where atoms from different subsets possibly overlap in time. Also, atoms may have complex temporal structures.

Various dictionaries have been used for audio signals so far. Dictionaries of windowed sinusoidal atoms have been used for speech modification in [18], then for audio coding in [19]. Complex expansions of these atoms, namely Gabor atoms, have been used for audio signal decompositions in [20] and applied to audio coding in [21], [22]. Other waveforms have been used: damped sinusoids [23], local cosines in addition to dyadic wavelet bases [24], *chirped* Gabor atoms [25], [26]. The latter are time-localized complex sinusoidal signals with linearly varying frequency defined by

$$g_{s,u,f,c}(t) = \frac{1}{\sqrt{s}}\, w\left(\frac{t-u}{s}\right) e^{2j\pi\left(f(t-u)+\frac{c}{2}(t-u)^2\right)} \tag{2}$$

where w is a finite-length window and s, u, f, c denote respectively the scale, time location, frequency and chirp rate parameters. In the remainder of this article, atoms are denoted by complex-valued signals, since in practice sparse decompositions of real-valued signals can involve pairs of atoms consisting of one complex-valued atom and its conjugate, as presented in [20], [23], [21]. Chirped Gabor atoms can efficiently represent most non-percussive musical sounds, which consist of sinusoidal partials with slowly-varying frequency and amplitude. However the resulting decompositions cannot be considered as mid-level representations, since they do not exhibit any pitch or timbre information.

Sparsity can be increased by designing an appropriate dictionary where the atoms exhibit more similarity with the analyzed signal. Obviously, this requires some prior knowledge about the signal. In this section, we define *instrument-specific harmonic* atoms and molecules based on the assumption that the analyzed signal involves instruments producing harmonic sinusoidal partials only and that the set of possible instruments is known.

A. Instrument-specific harmonic atoms

We define a harmonic atom as a sum of M windowed sinusoidal partials at harmonic frequencies with constant amplitudes but linearly varying fundamental frequency. Using chirped Gabor atoms to represent the partials, each harmonic atom is expressed as

$$h_{s,u,f_0,c_0,A,\Phi}(t) = \sum_{m=1}^{M} a_m \, e^{j\phi_m} g_{s,u,m\times f_0,m\times c_0}(t) \tag{3}$$

where s is the scale parameter, u the time location, f_0 the fundamental frequency, c_0 the fundamental chirp rate, $A = \{a_m\}_{m=1\ldots M}$ the vector of partial amplitudes and $\Phi = \{\phi_m\}_{m=1\ldots M}$ the vector of partial phases. The number of partials M is defined from f_0 so that the frequency $M \times f_0$ of the uppermost partial is just below the Nyquist frequency, with a maximum of 30 partials. In addition to the definition of harmonic atoms in [27], the proposed definition takes into account possible frequency variations using a chirp parameter and assumes that the partial amplitudes are fixed *a priori* instead of being determined from the analyzed signal. Also, the partial amplitudes satisfy the constraint

$$\sum_{m=1}^{M} a_m^2 = 1 \tag{4}$$

so that the resulting atoms have unit energy. This condition is fulfilled if we assume that f_0 is large enough so that the Gabor atoms $g_{s,u,m\times f_0,m\times c_0}(t)$ are pairwise orthogonal for different values of m. As mentioned in [27], the *quasi-orthogonality* of the partials for an atom of fundamental frequency f_0 depends on the scale s and the window w. We only consider the quasi-orthogonality of flat **(**$c = 0$**)**

harmonic atoms here since the search step in Section III-B is performed only on such atoms; the chirp rates are determined by a subsequent parameter tuning step as will be explained. In fact, for the lowest considered fundamental frequency, the modulus of the inner product between two consecutive partials of a flat harmonic atom is $|\langle g_{s,u,m\times f_0,0}, g_{s,u,(m+1)\times f_0,0}\rangle| \simeq 0.0261$.

The most distinctive feature of the proposed model is that the vector of partial amplitudes A is learned on isolated notes, so as to represent a single instrument i among all instruments possibly present in the polyphonic signal. More precisely, the frequency range of each instrument i is partitioned into several *pitch classes* p and each vector A is associated with a single instrument/pitch class $\mathcal{C}_{i,p}$. Each class $\mathcal{C}_{i,p}$ may contain several amplitude vectors denoted by $A_{i,p,k}$, with $k = 1 \dots K$. The learning of these vectors is detailed in Section V.

In addition to providing explicit instrument labels, instrument-specific amplitude vectors also potentially increase the accuracy of the representation by better discriminating instruments playing at the same time, as shown previously with different models for score transcription, source separation and instrument recognition [28], [29], [30]. The proposed model hence shares similar principles to template-based instrument recognition algorithms [28].

B. Instrument-specific harmonic molecules

While they help to describe instruments separately, instrument-specific harmonic atoms are time-localized and therefore do not capture long-term temporal content. A more informative mid-level representation may be obtained by replacing the *atomic* decomposition (1) by a better structured decomposition.

To this aim, we propose to decompose the signal as a set of instrument-specific harmonic molecules, where each molecule $\mathcal{M}$ is a group of instrument-specific harmonic atoms $\{h_\lambda(t)\}_{\lambda\in\mathcal{M}}$ satisfying the following constraints:

- the atoms span a range of time locations u, with exactly one atom per location,
- all atoms have the same instrument label i,
- the log-variation of fundamental frequency between any two consecutive atoms is bounded by a threshold D

$$|\log f_{0_{\lambda+1}} - \log f_{0_\lambda}| < D. \tag{5}$$

Figure 1 displays some example molecules modeling a solo flute signal. It can be observed that the fundamental frequencies, the weights and the fundamental chirp rates of the atoms vary over time within each molecule.

The two following sections are dedicated to the extraction of isolated atoms (III) and molecules (IV).

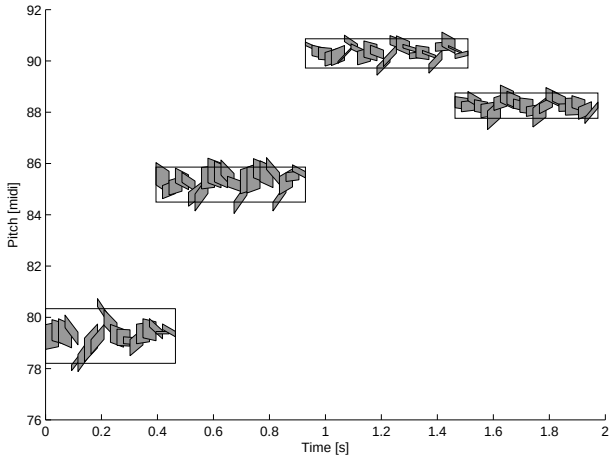


Fig. 1. Representation of a solo flute signal as a collection of harmonic molecules. Each atom is represented by a parallelogram centered at its time-pitch coordinates (u, f_0), whose width, height and inclination are respectively proportional to its scale s, weight α_λ and fundamental chirp rate c_0. Each molecule is depicted as a rectangle covering several atoms.

III. Extraction of prominent atoms

Given the two models of music signals defined above, the goal is now to decompose a signal using either one of these models at a reasonable computational cost. We concentrate in this section on the atomic decomposition (1) using the flat (constant f_0) atomic model described in Section II-A.

Many sparse decomposition techniques have been proposed in the literature. As in [20], we consider a given sparse decomposition to be optimal when it results in the best SRR among all decompositions with the same number of atoms. The Matching Pursuit (MP) algorithm, introduced in [20], extracts the atoms iteratively in order to maximize the SRR at each iteration. It is therefore only optimal at every step and not globally. In practical cases, this algorithm has been shown to provide near-optimal decompositions at small computational cost on standard dictionaries. However, it cannot be applied to a dictionary of harmonic atoms with fine resolution for each parameter, since the large number of parameters per atom would result in an extremely large dictionary. Thus we propose a modified MP algorithm, where each iteration consists of selecting the best atom from a dictionary with coarse resolution and tuning some of the parameters of this atom to maximize the SRR, as in [20]. Such an approach may be related to *weak* matching pursuit [31], that consists in selecting an atom that may not be optimal but whose modulus of its inner product with the signal is within close bounds of the optimal. In our case, such bounds cannot be computed in a straightforward manner. The asymptotic convergence towards zero of the proposed algorithms is not proven, which in practice is not a problem since we stop the algorithm

after few iterations.

A. Sampling of the dictionary

The resolution of the dictionary $\mathcal{D}$ is not imposed by the algorithm and can be chosen so as to achieve a suitable tradeoff between SRR and computational cost. For the applications in Section VI, the harmonic atoms are generated from a Hann window w and their discretized parameters s, u, f_0, c_0 and A are sampled as follows, assuming a sampling frequency of 22.05 kHz:

- the scale s is set to a single value, corresponding to a duration of 1024 samples (46 ms),
- the time location u is set to equally spaced frames, with a step Δu of 512 samples (23 ms),
- the fundamental frequency f_0 is logarithmically sampled, with a step $\Delta \log f_0$ of $\log(2)/60$ ($1/10$ tone),
- the fundamental chirp rate c_0 is set to 0,
- the vector of partial amplitudes A is one of the vectors $\{A_{i,p,k}\}_{k=1\ldots K}$ for the instrument i and the pitch class p that is the closest to f_0.

The logarithmic sampling of f_0 contrasts with the linear sampling used in [27] and is a natural choice for western music. Additional scales s could be chosen for applications requiring a high resynthesis quality, such as audio coding.

As proposed in [20], [23], the vector of partial phases Φ is not discretized, but computed from the data as a function of the other parameters in order to maximize the SRR

$$e^{j\phi_m} = \frac{\langle x, g_{s,u,m\times f_0,m\times c_0}\rangle}{|\langle x, g_{s,u,m\times f_0,m\times c_0}\rangle|} \tag{6}$$

where the inner product between two signals is defined by $\langle x, y\rangle = \sum_{t=1}^{T} x(t)\,\overline{y}(t)$.

B. Modified MP algorithm

The modified MP algorithm involves the following steps, as illustrated in Figure 2:

1) the inner products $\langle x, h_\lambda\rangle$ between the signal and all the atoms h_λ of the dictionary $\mathcal{D}$ are computed,
2) the atom h_e that gives the largest absolute inner product is selected

$$h_e = \arg\max_{h_\lambda\in\mathcal{D}} |\langle x, h_\lambda\rangle|, \tag{7}$$

3) the fundamental frequency f_0, the fundamental chirp rate c_0 and the partial phases Φ of this atom are tuned in order to maximize the SRR with s, u and A fixed. The optimization is performed

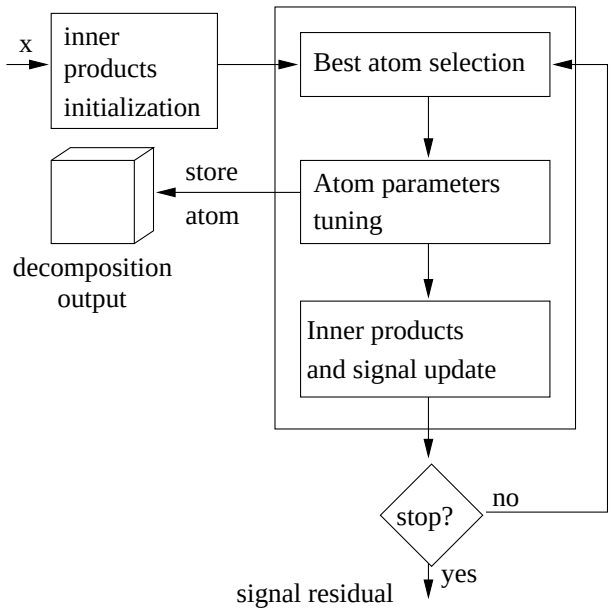


Fig. 2. Flow chart of the modified MP algorithm for the extraction of prominent harmonic atoms.

under the constraint that f_0 lies between the two neighboring bins of the fundamental frequency grid. Once done, the atom parameters and weight ($\alpha_e = \langle x, h_e \rangle$) are stored.

4) the tuned atom is subtracted from the signal, and the inner products $\langle x, h_\lambda \rangle$ are updated on the residual for all atoms h_λ temporally overlapping with h_e. The algorithm is iterated from step 2) until a target SRR has been achieved or a given number of atoms has been extracted. This atom tuning scheme leads to significant additional computational cost since the efficient inner product update described in [20] cannot be implemented here.

Parameter tuning is conducted using a conjugate gradient algorithm detailed in Appendix I.

We emphasize that this algorithm is applied to the whole signal as in [20], [32], as opposed to a number of approaches ([21], [22]) employing MP on a frame-by-frame basis.

IV. Extraction of prominent molecules

The modified MP algorithm presented in section III extracts atoms that exhibit significant harmonic structures in the analyzed signal, but that are independently extracted, with no explicit link between each other. One could perform a clustering step after atomic decompositions by grouping neighboring atoms into molecules. However, such a procedure would not lead to optimal molecules: the weights of the individual atoms are optimized independently from each other while the atoms are not orthogonal. This

typically occurs when a long signal structure is analyzed with short atoms, for example a long sinusoid with short Gabor atoms, as pointed in [33]. Indeed, the early extracted atoms catch a large part of the signal energy, while the subsequent ones are extracted to "fill the holes". In this case, the energy of the atoms may not follow the instantaneous energy of the analyzed structure. To overcome this issue, molecules are here directly extracted in the iterative process, leading to a representation where atoms have their parameters tuned to better fit the underlying music notes.

A molecule $\mathcal{M}$ is a set of neighboring atoms h_λ with their respective weights α_λ. The corresponding waveform μ is a linear combination of atoms:

$$\mu(t) = \sum_{\lambda \in \mathcal{M}} \alpha_\lambda h_\lambda(t) \tag{8}$$

In our case, the atoms h_λ within a molecule follow the constraints that have been defined in II-B. To stay consistent with the MP framework, finding the optimal molecule, in the least square sense, consists in finding the combination of weighted atoms $\mathcal{M}$ that maximizes $|\langle x, \mu \rangle|$ with the constraint $|\langle \mu, \mu \rangle| = 1$.

Given the atoms of one molecule and with no constraint set on the norm of μ, the optimal weights α_{0_λ} are computed with an orthogonal projection using the Gram determinant[1]:

$$\alpha_{0_i} = \frac{G(h_1, ..., h_{\lambda-1}, x, h_{\lambda+1}, ..., h_n)}{G(h_1, ..., h_{\lambda-1}, h_\lambda, h_{\lambda+1}, ..., h_n)} \tag{9}$$

If the constraint is set, the optimal weight vector (α_λ) is colinear to (α_{0_λ}):

$$\alpha_\lambda = \frac{\alpha_{0_\lambda}}{(\sum_{k \in \mathcal{M}} |\alpha_{0_k}|^2)^{1/2}} \tag{10}$$

Thus, the modulus of the inner product between the signal and the molecule signal, which we call the total weight δ_{opt} of the molecule $\mathcal{M}$, is:

$$\delta_{opt}(\mathcal{M}) = |\langle x, \mu \rangle| = \left| \frac{\sum_{\lambda \in \mathcal{M}} \overline{\alpha_{0_\lambda}} \langle x, h_\lambda \rangle}{(\sum_{\lambda \in \mathcal{M}} |\alpha_{0_\lambda}|^2)^{1/2}} \right| \tag{11}$$

The computation of the orthogonal projection of x on every set of atoms $\mathcal{M}$ to get the α_{0_λ} coefficients would be very costly. Thus, an additive structure based on the inner products between the signal and the individual atoms is desirable in order to facilitate a dynamic programming scheme. A heuristic weight δ is thus chosen *a priori* to estimate the best molecule (i. e. maximizing δ_{opt}):

$$\delta(\mathcal{M}) = \left(\sum_{\lambda \in \mathcal{M}} |\langle x, h_\lambda \rangle|^2 \right)^{1/2} \tag{12}$$

[1]The Gram determinant $G(x_1, x_2, \ldots, x_n)$ is the determinant of the Gram matrix defined by its elements $G_{i,j} = \langle x_i, x_j \rangle$.

This is the exact weight of the molecule if the atoms h_λ are orthogonal ($\alpha_{0_\lambda} = \langle x, h_\lambda \rangle$). This is clearly not the case in our study because the time-frequency supports of the atoms overlap. However this has little effect on the choice of the best molecule, since the ratio between $\delta_{opt}(\mathcal{M})$ and $\delta(\mathcal{M})$ is typically similar for the top candidate molecules. Nevertheless it is worth noting that optimizing with respect to δ would lead to the selection of molecules that are the longest possible, since adding any atom to a molecule increases its weight δ. We address this issue by introducing a two-step approach, each involving dynamic programing. First, a search time interval is delimited using the time support of the best molecule containing a preselected *seed atom* (equivalent to the best atom that is selected in the atomic algorithm, described in Section III). This support is obtained by extending the molecule until the atoms aggregated at the extremities fall below a predefined energy threshold. Second, the estimation of the best molecule is performed: it consists in searching the best atom path spanning this time interval. This two-step approach is also motivated by the difficulty to compare paths with different time supports. Once the best path has been found, the atom parameters are tuned and the atom weights are computed in order to maximize the SRR.

The whole iterative decomposition algorithm is depicted in Figure 3. The first two steps of the algorithm (namely inner products initialization and selection of the seed atom) are similar to those of the atomic approach. The subsequent steps are detailed below.

A. Search for the best atom path

The selection of a molecule $\mathcal{M}$ is equivalent to the search of the best path $\mathcal{P}$ on several instrument-specific time-pitch grids. These grids are built as follows: each node of the grid for instrument i is indexed by its discrete time location u and fundamental frequency f_0, as defined in Section III-A. It also carries a value $G_i(u, f_0)$ which is the maximum of the squared absolute inner products $|\langle x, h_\lambda \rangle|^2$ between the signal and the atoms h_λ of parameters u and f_0 over all the vectors of partial amplitudes $\{A_{i,p,k}\}_{k=1\ldots K}$ for the instrument i and the pitch class p that is the closest to f_0. Hence the weight of a path is defined by:

$$\delta(\mathcal{P}) = \left(\sum_{(u,f_0)\in\mathcal{P}} G_i(u, f_0) \right)^{1/2} \tag{13}$$

The node corresponding to the seed atom is called the *seed node*.

The search for the best atom path involves two steps. First, a single time search interval is delimited for all instruments using a Viterbi algorithm [34] on a restricted region. Then it is used again to find the best path within this interval for each instrument.

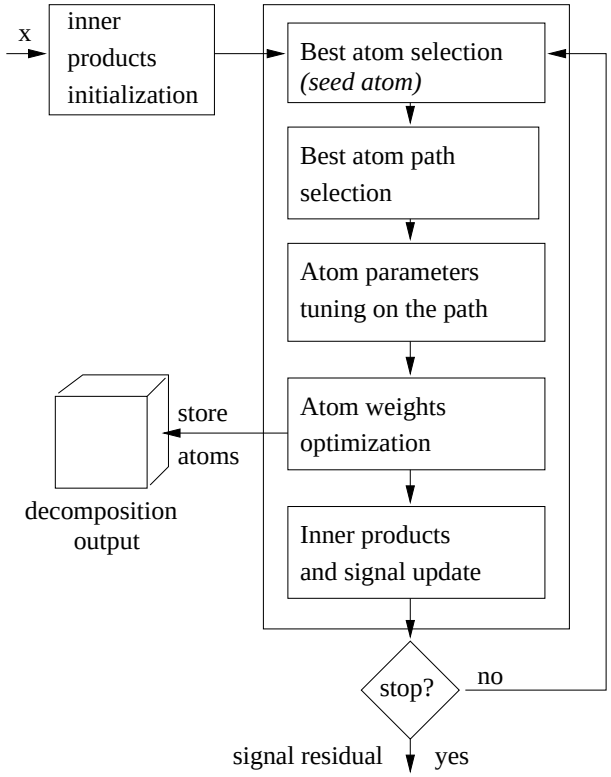


Fig. 3. Flow chart of the proposed algorithm for the extraction of prominent harmonic molecules.

1) The Viterbi algorithm: Suppose that a path is searched from an initial time u_0 towards increasing time (forward path search). Considering an instrument grid G_i, at time u and pitch f_0, the Viterbi algorithm is based on the principle:

$$\delta(u, f_0) = \max_{f_0' \in \mathcal{A}(f_0)} \delta(u-1, f_0') + G(u, f_0), \tag{14}$$

where $\mathcal{A}(f_0)$ is the set of pitch bins that can reach the pitch bin f_0, subject to the condition (5). Practically, the best path is constructed iteratively from the initial time to the final time, keeping track of the intermediate best paths along the search. Once a stopping condition is reached, a backtracking operation gives the best path. This algorithm has a complexity equal to $O((D/(\Delta \log f_0))^2 \times U^3)$, where U is the number of time bins of the path. Note that the total number of fundamental frequency bins N_{f_0} does not affect complexity. Indeed the search region is "triangular", so that the number of considered fundamental frequencies is proportional to the path duration. This algorithm is applied for the two steps described below.

2) Delimiting the time search interval: The search interval is delimited by computing a backward path and a forward path starting from the seed node. The algorithms for building these two paths are strictly symmetric. Only the algorithm for the forward path is detailed below.

The path that gives the largest weight under the constraints set in Section II-B can be estimated via the Viterbi algorithm described above, as illustrated in Figure 4. A transition is allowed between two successive atoms when their fundamental frequencies satisfy the constraint (5), resulting in a triangle-shaped pitch search region. The forward limit $u_{\max}$ ($u_{\min}$ for the backward limit) of the time search interval is attained when the value $G_i(u, f_0)$ of the last node of the current best path becomes lower than a threshold $A_{\mathcal{M}}$ defined as follows:

$$A_{\mathcal{M}} = \max\{\mu_0 \alpha_0^2, \mu_e \alpha_e^2\} \tag{15}$$

where $\alpha_0 = |\langle x, h_0 \rangle|$ is the weight of the first seed atom h_0 selected in the entire decomposition, $\alpha_e = |\langle x, h_e \rangle|$ is the weight of the seed atom h_e of the current molecule. μ_0 and μ_e are fixed ratios. The term $\mu_0 \alpha_0^2$ is a global energy threshold that prevents the selection of background noise atoms. μ_0 is typically chosen so that this threshold lies slightly below the energy threshold corresponding to the target SRR. The term $\mu_e \alpha_e^2$ introduces an adaptive energy threshold for each molecule, which avoids the selection of atoms belonging to subsequent or previous notes or to reverberation. Note that μ_e must be larger than μ_0, otherwise it has no effect on $A_{\mathcal{M}}$ because α_0 is almost always larger than α_e. Typical values for μ_0 and μ_e are given in Section VI.

3) Estimation of the best path: Once the time search interval has been determined, the Viterbi algorithm is applied for each instrument i on the rectangular grid delimited by the search interval $[u_{\min}, u_{\max}]$ and the whole instrument pitch range, this time without constraining the initial and final nodes. One path $\mathcal{P}_i$ per instrument i is thus obtained and the path with the largest weight is finally selected. Note that the initial seed atom is not used anymore and may not be included in the final molecule.

B. Tuning of atom parameters

In order to improve the parameter resolution of the atoms of the selected molecule compared to that of the dictionary $\mathcal{D}$, the parameters f_0, c_0 and Φ are tuned for each atom at a time so as to maximize the SRR under the constraint that f_0 stays between the two neighboring bins of the fundamental frequency grid, while keeping the parameters of other atoms fixed. This tuning is conducted using a conjugate gradient algorithm [35], as described in Appendix I.

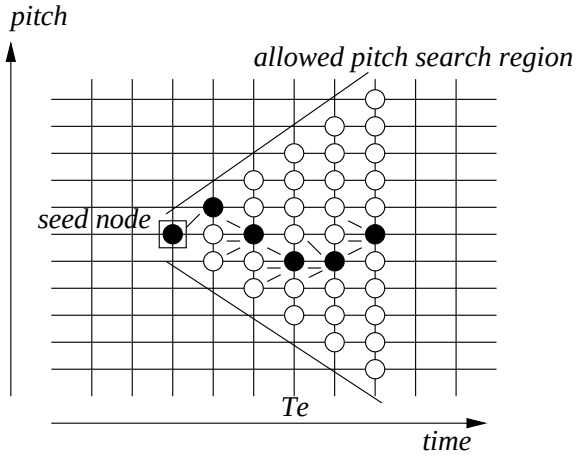


Fig. 4. Selection of the forward path. The frequency log-variation threshold D is assumed to be equal to the frequency discretization step and the pitch search region is delimited by the two diagonal lines. The current best path is composed of black atoms. The little segments before the black atoms show the possible previous atoms for each black atom.

C. Computation of the atom weights

As pointed in Section IV, the atoms contained in a molecule are not orthogonal. Thus, the optimal weights α_{0_λ} are computed *a posteriori* using (9) as in [23].

D. Signal and inner products update

Once the optimal weights are computed, the molecule is subtracted from the signal by subtracting each atom h_λ scaled by the corresponding weight α_λ.

The inner products $\langle x, h_\lambda \rangle$ are updated on the residual for all atoms h_λ temporally overlapping with at least one atom of the molecule. The algorithm is then iterated from the seed atom selection until no remaining atom satisfies constraint (15), a target SRR has been achieved, or a predefined number of atoms has been extracted. These stopping criteria are chosen so as to avoid the extraction of spurious low energy molecules.

E. Complexity and scalability

The computational load is dominated by the update of the inner products and the parameter tuning for each algorithm. The following discussion evaluates the complexity of a single iteration of the algorithm. We recall that D is the maximum log-fundamental frequency deviation between two consecutive atoms in a molecule, M the number of partials of an instrument-specific harmonic atom and K the number of

atoms per $\mathcal{C}_{ip}$ set. With U the number of time steps of the atom path and N_s the scale in number of samples, I the number of instruments in the dictionary and N_{f0} the number of fundamental frequencies, the load of the Viterbi algorithm is $O(I \times (D/\Delta \log f_0)^2 \times U^3)$ and the atom weights optimization $O(U^2 \times N_s)$. As we will see, they can be neglected with regard to the following operations. Given an iteration of the algorithm, the computation of inner products involves first a computation of the inner products between the signal and Gabor atoms ($O(M \times N_{f0} \times N_s \times U)$) where N_s is the scale in number of samples. Note that for the first iteration, U is the total number of time frames because all the inner products must be initialized. Then, the inner products between the resulting projections $|\langle x, g \rangle|$ and the partial amplitudes vectors A are computed with this complexity: $O(M \times K \times I \times N_{f0} \times U)$. The parameter tuning has the following complexity: $O(M \times N_s \times N_{it} \times U)$, where N_{it} is the number of iteration of the gradient algorithm.

In Section VI, the representations are evaluated with a set of five instruments, considering that each of the instruments in the ensemble plays a maximum of one note at a time, an assumption that is verified in the large majority of playing conditions for the five considered instruments. With the chosen parameters presented in Section VI, each algorithm takes approximately one minute to process one second of signal with the current Matlab implementation on a machine equipped with a Pentium IV 3 GHz. The computational load is mainly devoted to the tuning of the parameters (over 50% of the load) and to the update of the inner products (about 30%). However, it must be mentioned that this computational load is needed only once for the decomposition algorithms: once performed, each of the post-processing procedures (see section VI) are extremely fast (a fraction of real time).

We assess the potential of the proposed representations for ensembles taken within a set of five monophonic instruments, which admittedly is a restricted context. These representations and the associated instrument recognition algorithms for solo performances would however be directly applicable to instruments playing chords. Indeed each chord could be decomposed at no additional computation load as a sum of atoms or molecules, each one corresponding to a different note. The proposed representations would also be applicable to a larger number of instruments I, e.g. 40 instead of 5. In this context, the contribution of the inner products with the A vectors would become prominent and increase the computational load linearly with the number of instruments I in the dictionary, which remains tractable. Moreover, hierarchical procedures [36] can be used to reduce the contribution of this load.

V. Learning the dictionary parameters

For the following experiments, the vectors of partial amplitudes $\{A_{i,p,k}\}_{k=1...K}$ are learned for each instrument/pitch class $\mathcal{C}_{i,p}$ on isolated notes from three databases: the RWC Musical Instrument Sound Database [37], IRCAM Studio On Line [38] and the University of Iowa Musical Instrument Samples [39]. We select five instruments that produce harmonic notes: oboe (Ob), clarinet (Cl), cello (Co), violin (Vl) and flute (Fl). While recently developed approaches involve unsupervised and data-driven methods to build dictionaries [40], the learning is here done in a supervised way: atoms are associated to a pitch and a label.

For each isolated note signal, the time frame with maximal energy is computed and all the subsequent time frames whose energy lies within a certain threshold of this maximum are selected. This relative threshold is set to a ratio of 0.05 in the following. The partial amplitudes are computed for each of these training frames by

$$a_m = \frac{|\langle x, g_{s,u,m\times f_0,m\times c_0}\rangle|}{\left(\sum_{m'=1}^{M} |\langle x, g_{s,u,m'\times f_0,m'\times c_0}\rangle|^2\right)^{1/2}} \tag{16}$$

where f_0 and c_0 are tuned in order to maximize the SRR on this frame, using the conjugate gradient algorithm described in Appendix I. The vector of amplitudes is then associated to the pitch class p that is the closest to f_0. The resulting number of vectors per instrument and per pitch class are indicated in Table I.

Instrument	Number of training frames	Average number per pitch
Ob	5912	169
Cl	9048	193
Co	13868	285
Vl	37749	700
Fl	13216	330

TABLE I

Total number of training time frames per instrument and average number per pitch class.

The size of the dictionary varies linearly as a function of the number of amplitude vectors. Since the number of vectors is too large to ensure computationally tractable decompositions, we choose to reduce the number of vectors by vector quantization: K amplitude vectors are kept for each class $\mathcal{C}_{i,p}$ using the k-means algorithm with the Euclidean distance. The choice of this distance is justified by the

SRR objective, as shown in Appendix II. This operation also helps avoiding overfitting by averaging the training data and removing outliers.

VI. Applications

In this section, we evaluate the potential of the proposed representations for music visualization and polyphonic instrument recognition. The number of atoms per instrument/pitch class is set to $K = 16$, the noise level ratios μ_0 and μ_e to 0.03 and 0.2 respectively, and the frequency log-variation threshold D to $\log(2)/60$ (1/10 tone). The atom parameters are discretized as specified in section III-A.

A. Music visualization

The proposed mid-level representations can be used to visualize short- or long-term harmonic content and orchestration by plotting the estimated atoms or molecules on a time-pitch plane with instrument-specific colors. The representations provide a simple solution to the task of joint pitch and instrument transcription for polyphonic music signals, while the majority of polyphonic transcription algorithms output pitch estimates only.

It is a common view that this task could be addressed instead by performing monophonic pitch transcription and instrument recognition on the outputs of a blind source separation algorithm, which should consist of a single instrument. However single-channel source separation algorithms typically rely on factorial Gaussian mixtures or hidden Markov models which exhibit exponential complexity with respect to the number of instruments, which makes them unusable so far for more than two instruments known a priori [29]. Another similar view is that this task could be solved by performing first polyphonic pitch transcription, then extracting each note via harmonicity-based source separation and applying monophonic instrument recognition to the separated note signals. However harmonicity does not provide sufficient information to reliably transcribe and separate notes sharing partials at the same frequencies, which typically results in erroneous instrument labels [30]. By contrast, our algorithms rely on timbre information at all steps, thus avoiding erroneous pitch estimates whose timbre does not correspond to any existing instrument. Also it has linear complexity with respect to the number of possible instruments, which is much lower than the complexity of previous algorithms [29], [28] based on the same idea.

Figure 5 displays the representations computed from a recorded 10-second flute and clarinet excerpts extracted from a commercial CD, with a target SRR of 15 dB or a maximal number of 250 atoms per second. The upper melodic line is played by the flute, the lower one by the clarinet.

The atomic decomposition provides a relevant representation of the music being played, as shown by comparison with the ground truth piano roll. However, some of the extracted atoms have a large fundamental chirp rate that does not correspond to the actual variation of fundamental frequency of the music notes: the parameter tuning of an individual atom is perturbed by the residual coming from the extraction of a neighboring atom.

The molecular decomposition seems close to what can be expected for a time-pitch music representation: notes appear quite clearly as localized patches, instruments are often well identified. Compared to the visualization of the output of the atomic decomposition, the aforementioned drawback is considerably reduced, and the frequency and amplitude modulations within the music notes are here clearly visible. The relevance of the representation lets us expect that this decomposition could be sufficient as a front-end for key finding [41] or melodic similarity assessment [42], and perhaps for polyphonic pitch transcription using efficient post-processing methods based on musicological rules [13].

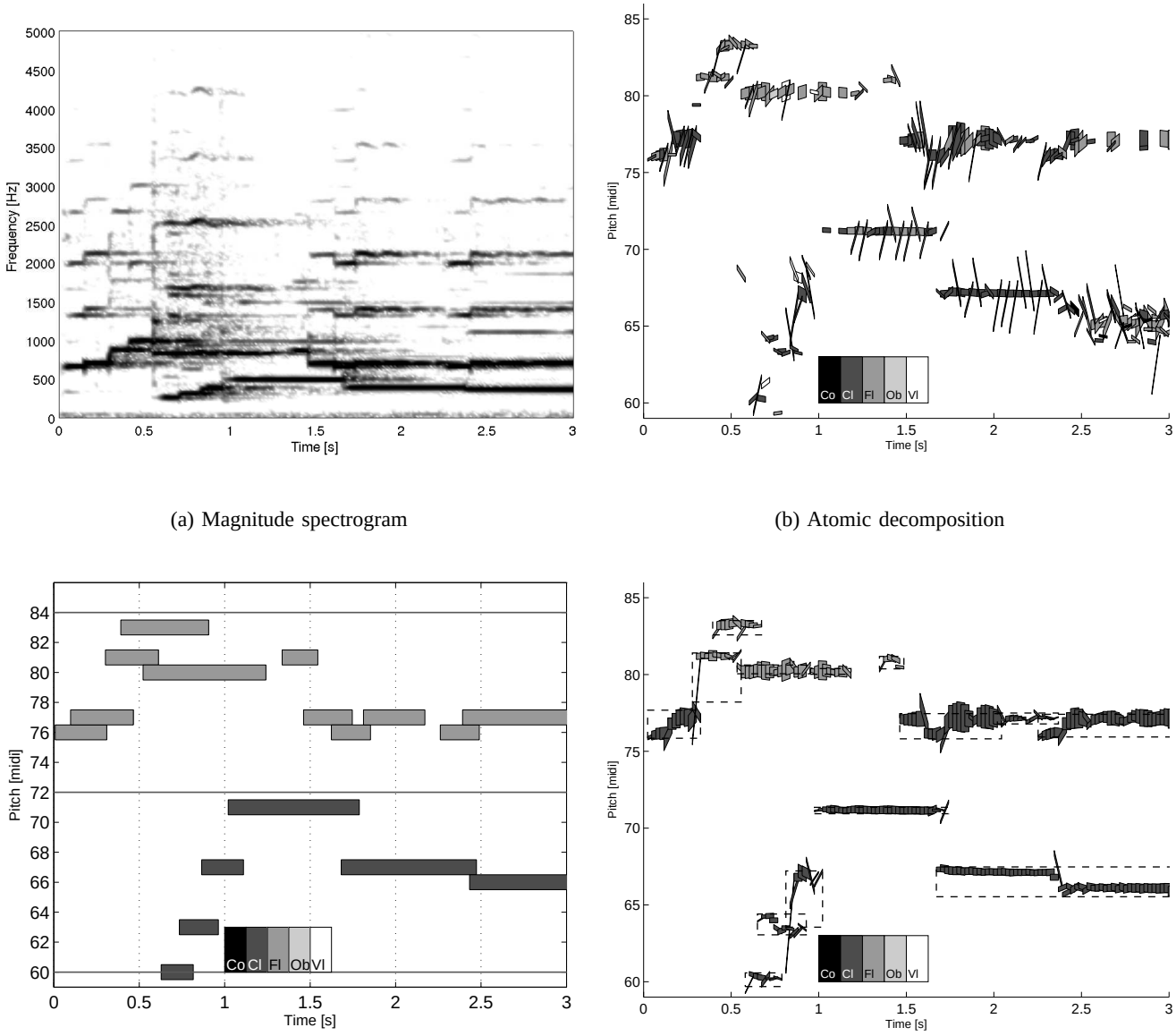
B. Solo musical instrument recognition

Musical instrument recognition on solo phrases has been handled in [43], [44], [45], [9] with "bag-of-frames" approaches. The performances are now close to what expert musicians can do. In [44], expert musicians showed an average performance of 67 % for the identification of 10-s excerpts among 27 instruments, while the complete system described in [46] reaches 70 % for a similar test case. These methods cannot be employed directly for multi-instrument music without learning appropriate models for every possible instrument combination.

The nature of the presented decomposition output suggests an application to musical instrument recognition, since each atom is associated with a specific instrument. Before validating this claim on polyphonic music signals, we evaluate the instrument discriminating power on solo excerpts as a benchmark. In this case, one way of identifying the underlying instrument waveform is to compute a score S_i for each instrument class i and to select the instrument with the largest score S_i. We propose to express this score as a function of the absolute weights $|\alpha_\lambda|$ of all the extracted atoms from this class:

$$S_i = \sum_{\lambda \in B_i} |\alpha_\lambda|^\gamma, \tag{17}$$

where B_i is the set of indexes of the atoms that come from instrument i, and to select the instrument with the largest score S_i. The γ coefficient is optimized to balance the importance of high- and low-energy atoms. On a development set whose contents are similar to the test set but come from different sources, the best classification accuracy was achieved for $\gamma = 0.2$



(a) Magnitude spectrogram

(b) Atomic decomposition

(c) Ground truth piano roll

(d) Molecular decomposition

Fig. 5. Visualization of a flute and clarinet duo, compared to the ground truth piano roll. Each atom is represented by a grayscale patch centered at its time-pitch coordinates (u, f_0), whose width, height and inclination are respectively proportional to its scale s, weight α_λ and chirp rate c_0. Each molecule is depicted as a dashed-line rectangle covering several atoms. The grayscale indicates the instrument associated with each atom.

The goal of this experiment on real solo phrases is to provide a benchmark of the performances of the algorithm on realistic signals. In other words, before applying it to polyphonic signals we have to check that it has good discrimination capabilities for the instruments waveforms. The decomposition algorithms are applied on a test database of 2-second solo excerpts, obtained by partitioning five tracks from different commercial CDs for each instrument, discarding silence intervals. The number of excerpts is shown in Table II. This yields 95% confidence intervals smaller than 0.1% on the measured recognition accuracies. However, it is worth noting that this confidence interval is rather optimistic since it relies on the independence of the test samples, whereas test samples coming from the same recording cannot reasonably be considered as independent. The algorithms are stopped when the SRR becomes larger than 10 dB or the number of extracted atoms reaches 100 atoms per second.

Instrument	Test sources	Duration
Ob	5	14’40”
Cl	5	13’38”
Co	5	12’7”
Vl	5	24’11”
Fl	5	15’56”

TABLE II

CONTENTS OF THE TEST DATABASE FOR SOLO INSTRUMENT RECOGNITION.

The classification results obtained from the atomic decomposition and the molecular decomposition are given in Table III in the form of a confusion matrix. The average recognition accuracy equals 77.3% for the atomic decomposition and 73.2 % for the molecular decomposition. Note that all instruments are often confused with the flute, which could be due to the fact that the flute exhibits some prototypical characteristics common to all instruments, as suggested in [4].

If a larger number of instruments is considered, for instance 40, the decomposition algorithm would still be tractable since the computation time is approximately linear with the number of instruments and that less atoms per pitch class can be kept (e.g. 8 instead of 16). However, the raw music instrument recognition results should drop, as for any music instrument recognition algorithm. In this case, the atoms would still align to the correct played pitches, but the instrument labels would not be reliable enough to derive a good annotation of the signal. Nevertheless music instrument recognition in a such open context would be possible for families of instruments (simple reed, double reed, bowed strings, brass...), whose

prototypical characteristics given by the physical production mode of the sounds can be caught in the partial amplitudes vectors.

%	Ob			Cl			Co			Vl			Fl		
Ob	**73.8**	**66.0**	***81.1***	3.6	3.6	*5.8*	4.9	7.4	*2.8*	2.9	5.8	*9.3*	14.9	17.2	*0.8*
Cl	0.4	0.4	*2.8*	**82.2**	**74.9**	***56.0***	0	2.2	*19.4*	1.3	2.2	*7.8*	16.0	20.2	*13.7*
Co	0	0.0	*0.0*	6.8	4.3	*1.2*	**81.5**	**81.6**	***88.1***	1.5	4.3	*8.0*	10.2	9.8	*2.7*
Vl	1.1	0.7	*2.3*	4.4	3.0	*2.3*	8.8	12.5	*2.1*	**62.9**	**56.1**	***87.1***	22.8	27.7	*6.2*
Fl	3	3.4	*1.7*	11.1	7.7	*4.0*	0	1.3	*16.2*	0	0.4	*2.6*	**86.0**	**87.1**	***75.5***
Algorithm	At.	Mol.	SVM	At.	Mol.	SVM	At.	Mol.	SVM	At.	Mol.	SVM	At.	Mol.	SVM

TABLE III

RESULTS OF SOLO INSTRUMENT RECOGNITION USING THE ATOMIC DECOMPOSITION (FIRST COLUMNS), THE MOLECULAR DECOMPOSITION (MIDDLE COLUMNS) AND THE SVM(MFCC)-BASED ALGORITHM (LINES: TESTED, COLUMNS: ESTIMATED).

The proposed classification systems exploit a reduced part of what constitutes musical timbre, namely the spectral envelope of the harmonic content. Hence they can be compared to standard solo instrument recognition algorithms exploiting spectral envelope features. We apply the algorithm described in [9] to 10 MFCC, which are known to represent the prominent characteristics of spectral envelopes. This algorithm uses Support Vector Machines (SVM) within a pairwise classification strategy. While it cannot be considered as a state-of-the-art system, it gives a good indication of what a very good classifier can do with widely approved features for timbre discrimination. It achieves an average recognition accuracy of 77.6%, when the SVM is trained on the same isolated note signals as in Section V. This score is only slightly higher than the ones reported above. It should be remarked that the confusions do not happen on the same instruments. For instance, the SVM(MFCC)-based system fails at identifying the Clarinet, while the Violin is the weak point of our algorithms. It must be remarked that the overall scores are lower than figures appearing in other works on music instrument recognition. It is mainly related to the differences between the training set, composed of isolated notes recorded in almost anechoic conditions, and the test set, made of real recordings with subsequent room effects, and sometimes double notes for string instruments. The adaptation of amplitude parameters learning on real recordings gives a track for investigations. Indeed, the similarity of training data and the test data is a critical aspect for the success of a classifier. Other experiments we have performed with the SVM-based classifier with more features

[46] have shown results increased by 10 % if the classifier is learned on solos. A similar increase of performances can be expected for our classifier, leading to results that can be a good basis for further processing.

C. Polyphonic musical instrument recognition

While as effective as standard feature-based approaches for solo musical instrument recognition, the proposed mid-level representations are naturally capable of handling polyphonic music. The experiment that will be described aim at showing that this task is possible without changing the learning step. By contrast, feature-based approaches would have to learn one classifier for every combination of instruments, which would quickly become prohibitive given the combinatorial explosion of the number of classes for which sufficient training data must be collected: $\binom{I+I_{active}-1}{I_{active}}$, where I_{active} is the number of instruments playing and I the number of possible instruments. Moreover, some features, such as inharmonic content features, cannot be robustly extracted in polyphonic signals anymore.

Polyphonic music instrument recognition is still an emerging topic, and very few studies involve extensive tests on real recordings. These approaches involve bag-of-frames techniques [5], template-based approaches [28], [47], [48], [49] or prior source separation [50]. Among all the listed works, some of them cannot be easily implemented, others have a computational complexity too high to be performed on the entire database. For example, [5] requires the definition of 20 classes (15 duos, 5 solos) for our instrument set, and if a pairwise strategy was applied as in VI-B, it would need the training of 190 classifiers which is nearly intractable. Some methods ([28], [47]) are based on Computational Auditory Scene Analysis and composed of different complex modules necessitating fine and dedicated tuning, and with no available public implementation. [48], [49], [50] state that the total number of instruments is known and that the involved instruments are different, moreover [48] remains extremely complex and cannot be applied on the entire database. Thus, the experiments will only be performed for the two algorithms that have been developed.

To show the potential of the approach, we provide polyphonic instrument recognition results on a database, even if they cannot be directly compared to the results another algorithm. The test database is composed of 2-second excerpts involving four instrument pairs: ClFl, CoFl, FlFl and CoVl. These excerpts are obtained by partitioning duo tracks from different commercial CDs. Note that, because the partitioning is unsupervised, some excerpts may contain one instrument only instead of two. The stopping criteria for the decomposition algorithms are a SRR of 15 dB or a maximal number of 250 atoms per second.

The following scoring method is chosen: each time frame is associated with one or two instrument labels by selecting the two atoms with the largest absolute weight $|\alpha_\lambda|$, or one atom only if there is only one in this frame. Then the label of the entire excerpt is decided by a majority vote, weighted by the sum of the absolute weights of the atoms in each frame. This method does not take any musicological knowledge into account, for example the separation of the melodic lines. Implementing a melodic line tracking in the time-pitch plane is left for further research, but definitely possible in this framework.

Three distinct accuracy scores are computed. The score A measures the recognition accuracy of the actual duo, or of a single instrument of the actual duo if only one is detected. For instance, if the ground truth label is CoFl, the correct detections are Co, CoFl and Fl. The score B counts a good detection when all the detected instruments belong to the actual duo. Considering the example above, the labels CoCo and FlFl are also accepted. Finally, the score C indicates the recognition accuracy of detecting at least one instrument of the actual duo. In our example, the labels CoVl, CoOb, CoCl, ClFl, ObFl and FlVl are added. The scores obtained using a random draw would equal 15%, 25% and 55% respectively for duos of different instruments (ClFl, CoFl and CoVl) and 10%, 10% and 30% respectively for duos of identical instruments (FlFl), considering all the labels with equal probability.

The scores obtained from the atomic decomposition and the molecular decomposition are presented in Tables IV and V respectively.

%	Number of excerpts	A	B	C
ClFl	200	55.0	78.0	97.0
CoFl	170	40.0	81.2	97.6
FlFl	29	48.3	48.3	93.4
CoVl	414	23.2	70.0	96.1
Overall	813	35.4	73.6	96.6

TABLE IV

RESULTS OF INSTRUMENT RECOGNITION ON DUOS USING THE ATOMIC DECOMPOSITION (A: ACTUAL DUO OR SOLO, B: PRESENT INSTRUMENTS ONLY, C: AT LEAST ONE PRESENT INSTRUMENT).

The molecular algorithm shows a better performance for the score A than the atomic algorithm, and slightly lower performances for scores B and C. It must be noted that the scores are computed on 2-second segments, which is a quite short period to take a decision. 10-second decisions give higher results, but in this case the number of evaluations does not lead to statistically meaningful scores. Here again, the

%	Number of excerpts	A	B	C
ClFl	200	58.0	87.0	98.0
CoFl	170	55.3	78.8	100.0
FlFl	29	69.0	69.0	89.7
CoVl	414	24.2	59.4	89.4
Overall	813	40.6	70.6	93.7

TABLE V

RESULTS OF INSTRUMENT RECOGNITION ON DUOS USING THE MOLECULAR DECOMPOSITION (A: ACTUAL DUO OR SOLO, B: PRESENT INSTRUMENTS ONLY, C: AT LEAST ONE PRESENT INSTRUMENT).

more structured decompositions coming from the molecular decomposition (see the example of duo on Figure 5) let us expect that the performances can be improved by using adequate post-processing, for example to split molecules that may contain several notes, and by constructing molecule-based features exhibiting amplitude and frequency modulations.

D. Remark and perspectives for music instrument recognition

For both experiments, on solo and duo performances, it is important to note that there are several ways to build the decision procedure. The procedure for the solo case involves the amplitudes of the atoms, and thus gives the most importance to the most energetic atoms, while the one in the duos case involves a frame-based decision weighted by the atom amplitudes, which rather emphasizes instruments that are playing on most of the time frames. In the solo case, preliminary experiments have shown that a weighted frame-based method performs worse than the presented amplitude-based method (7 to 10 points less for the overall score), but better than a purely frame-based method (one hard decision taken per time frame). On the opposite, in the duo case, the weighted frame-based method performs as well as a purely frame-based method.

As a perspective, the balance between the structure-based decisions (atoms or molecules) versus frame-based decisions is a subject of study, but beyond the scope of this paper.

VII. CONCLUSION

We have introduced in this paper a novel mid-level representation of music signals, based on the decomposition of a signal into a small number of harmonic atoms or molecules bearing explicit musical

instrument and pitch labels. The key feature of this system is that the parameters of each atom are learned from real recordings of isolated notes. Deterministic greedy algorithms derived from matching pursuit can be used to extract these decompositions from real polyphonic recordings, with good accuracy and reasonable complexity.

These decompositions, although very compact and prone to some transcription errors (as any such system without high-level musicological models), retain some of the most salient features of the audio data. In fact, this object-based decomposition could be used in numerous applications such as object-based audio coding, sound indexing and modification. Such post-processing on the decomposition outputs will have to be evaluated compared to task-specific audio processing.

In this paper, the potential of these representations has been thoroughly demonstrated on the task of automatic musical instrument recognition. On monophonic excerpts, the proposed algorithms obtained have nearly equivalent performances than a standard MFCC feature-based approach. Furthermore, the full benefits come when considering polyphonic music, where a basic post-processing method leads to encouraging results on realistic signals. Future work will be dedicated to a number of possible improvements, and in particular to the extension of the dictionary to include additional atoms capturing more diverse timbre information such as intrinsic amplitude and frequency modulations, transient characteristics or noise components, as well as perceptual criteria.

ACKNOWLEDGMENT

This work started when Pierre Leveau visited the Centre for Digital Music at Queen Mary University of London, where Emmanuel Vincent was a research assistant. The authors wish to thank other researchers of the Centre for Digital Music for fruitful discussions: Mark Plumbley, Mark Sandler, Juan Bello and Thomas Blumensath. Many thanks to Rémi Gribonval for a very interesting piece of advice for this work and to Slim Essid for providing us his automatic instrument recognition algorithm.

APPENDIX I

The tuning of the parameters f_0 and c_0 is performed via a conjugate gradient algorithm, where the partial phases Φ are reestimated at each iteration using (6). The maximization of the SRR is equivalent to the maximization of the square inner product between the atom and the signal

$$\mathcal{J} = |\langle x, h_\lambda \rangle|^2. \tag{18}$$

Assuming that the partial atoms $g_{s,u,m\times f_0,m\times c_0}(t)$ (denoted $g_m(t)$ for conciseness) are pairwise orthogonal, this gives

$$\mathcal{J} = \sum_{m=1}^{M} a_m^2 . |\langle x, g_m \rangle|^2. \tag{19}$$

The gradient of this quantity is defined by

$$\nabla \mathcal{J} = \left[\frac{\partial \mathcal{J}}{\partial f} \frac{\partial \mathcal{J}}{\partial c} \right]^T \tag{20}$$

Denoting by $\otimes$ the sample-wise product of two signals, and with $\frac{\partial g_m}{\partial f_0} = 2j\pi m(t-u) \otimes g_m$ and $\frac{\partial g_m}{\partial c_0} = 2j\pi m \frac{(t-u)^2}{2} \otimes g_m$, we obtain

$$\frac{\partial \mathcal{J}}{\partial f_0} = 2 \sum_{m=1}^{M} a_m^2 . \Re \left(\langle x, 2j\pi m(t-u) \otimes g_m \rangle \overline{\langle x, g_m \rangle} \right), \tag{21}$$

$$\frac{\partial \mathcal{J}}{\partial c_0} = 2 \sum_{m=1}^{M} a_m^2 . \Re \left(\left\langle x, 2j\pi m \frac{(t-u)^2}{2} \otimes g_m \right\rangle \overline{\langle x, g_m \rangle} \right). \tag{22}$$

This can also be written

$$\nabla \mathcal{J} = -4\pi \sum_{m=1}^{M} a_m^2 . m \times \Im \left(\left\langle \begin{bmatrix} t-u \\ \frac{(t-u)^2}{2} \end{bmatrix} \otimes x, g_m \right\rangle \overline{\langle x, g_m \rangle} \right) \tag{23}$$

A fast computation of this gradient can be achieved by storing the signals $t \times x(t)$ and $t^2 \times x(t)$.

APPENDIX II

If we detail the atom selection step of the atomic decomposition algorithm, it can be remarked that it acts as an adaptive classifier in the space of partial amplitudes vectors. The atom selection step is defined by (7) where

$$\langle x, h_\lambda \rangle = \left\langle x, \sum_{m=1}^{M} a_m \, e^{j\phi_m} g_{s,u,m\times f_0,0} \right\rangle \tag{24}$$

$$= \sum_{m=1}^{M} a_m \, e^{-j\phi_m} \langle x, g_{s,u,m\times f_0,0} \rangle \tag{25}$$

According to the definition of the estimated partial phases in (6), we get

$$\langle x, h_\lambda \rangle = \sum_{m=1}^{M} a_m \, |\langle x, g_{s,u,m\times f_0,0} \rangle| \tag{26}$$

which can be written using a normalization factor C as

$$\langle x, h_\lambda \rangle = C \sum_{m=1}^{M} a_m \, b_m \tag{27}$$

where

$$C = \left(\sum_{m=1}^{M} |\langle x, g_{s,u,m\times f_0,0} \rangle|^2 \right)^{1/2}, \tag{28}$$

$$b_m = \frac{1}{C} |\langle x, g_{s,u,m\times f_0,0} \rangle|. \tag{29}$$

Denoting by $B = \{b_m\}_{m=1...M}$ the vector of observed partial amplitudes satisfying $\sum_{m=1}^{M} b_m^2 = 1$, we finally obtain

$$|\langle x, h \rangle| = C \langle A, B \rangle. \tag{30}$$

This shows that the absolute inner product between the atom and the signal is the product of two factors: the square root C of the energy of the signal at multiples of the fundamental frequency f_0 and the inner product between the normalized amplitude vector A from the dictionary and the normalized amplitude vector B observed on the signal. The latter is also equal to one minus half of the Euclidean distance between A and B, which justifies the use of this distance for the clustering of amplitude vectors.

REFERENCES

[1] B. T. Logan, "Mel frequency cepstral coefficients for music modeling," in *Proc. of Int. Symp. on Music Information Retrieval (ISMIR)*, 2000.

[2] M. A. Bartsch and G. H. Wakefield, "To catch a chorus: using chroma-based representations for audio thumbnailing," in *Proc. of IEEE Int. Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2001, pp. 15–18.

[3] ISO/IEC 15938-4:2002, "Information technology - Multimedia content description interface - Part 4: Audio," 2002.

[4] J.-J. Aucouturier, "Ten Experiments on the Modelling of Polyphonic Timbre," Ph.D. dissertation, University Pierre et Marie Curie, Paris, France, 2006.

[5] S. Essid, G. Richard, and B. David, "Instrument recognition in polyphonic music based on automatic taxonomies," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 14, no. 1, pp. 68–80, January 2006.

[6] O. Izmirli, "Tonal similarity from audio using a template based attractor model," in *Proc. of Int. Conf. on Music Information Retrieval (ISMIR)*, 2005, pp. 540–545.

[7] M. Levy, M. B. Sandler, and M. A. Casey, "Extraction of high-level musical structure from audio data and its application to thumbnail generation," in *Proc. of IEEE Int. Conf. on Audio, Speech and Signal Processing (ICASSP)*, 2006, pp. V–13–16.

[8] G. Tzanetakis and P. Cook, "Musical Genre Classification of Audio Signals," *IEEE Trans. on Speech and Audio Processing*, vol. 10, no. 5, p. 293, July 2002.

[9] S. Essid, G. Richard, and B. David, "Musical instrument recognition by pairwise classification strategies," *IEEE Trans. on Audio, Speech and Language Processing*, July 2006.

[10] C. McKay and I. Fujinaga, "Automatic genre classification using large high-level musical feature sets," in *Proc. of Int. Conf. on Music Information Retrieval (ISMIR)*, 2004, pp. 525–530.

[11] D. Temperley, "A Bayesian approach to key finding," in *Music and Artificial Intelligence*, C. Anagnostopoulou, M. Ferrand, and A. Smaill, Eds. Springer, Berlin, Germany, 2002, pp. 195–206.

[12] M. Grachten, J. L. Arcos, and R. López de Mántaras, "Melodic similarity: looking for a good abstraction level," in *Proc. of Int. Conf. on Music Information Retrieval (ISMIR)*, 2004, pp. 210–215.

[13] A. Klapuri and M. Davy, Eds., *Signal Processing Methods for Music Transcription*. Springer, New York, NY, 2006.

[14] D. P. W. Ellis and D. F. Rosenthal, "Mid-level representations for computational auditory scene analysis," in *Computational auditory scene analysis*, D. F. Rosenthal and H. G. Okuno, Eds. Lawrence Erlbaum Associates, Mahwah, NJ, 1998, pp. 257–272.

[15] S. Dixon, F. Gouyon, and G. Widmer, "Towards characterization of music via rhythmic patterns," in *Proc. of Int. Conf. on Music Information Retrieval (ISMIR)*, 2004, pp. 509–516.

[16] J. P. Bello and J. Pickens, "A robust mid-level representation for harmonic content in music signals," in *Proc. of Int. Conf. on Music Information Retrieval (ISMIR)*, 2005, pp. 304–311.

[17] T. Kitahara, M. Goto, K. Komatani, T. Ogata, and G. Okuno, "Instrogram: A new musical instrument recognition technique without using onset detection nor f0 estimation," in *Proc. of IEEE Int. Conf. on Audio, Speech and Signal Processing (ICASSP)*, vol. 5, 2006, pp. 229–232.

[18] E. B. George and M. J. T. Smith, "Analysis-by-synthesis/overlap-add sinusoidal modeling applied to the analysis and synthesis of musical tones," *Journal of the Audio Engineering Society*, vol. 40, no. 6, pp. 497–516, June 1992.

[19] K. Vos, R. Vafin, R. Heusdens, and W. Kleijn, "High-quality consistent analysis-synthesis in sinusoidal coding," *17th Audio Engineering Society International Conference*, pp. 244–250, September 1999.

[20] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, December 1993.

[21] M. M. Goodwin, "Multiscale overlap-add sinusoïdal modeling using matching pursuit and refinements," in *Proc. of IEEE Int. Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, IEEE, Ed., October 2001, pp. 207–210.

[22] R. Heusdens, R. Vafin, and W. B. Kleijn, "Sinusoidal modeling using psychoacoustic-adaptive matching pursuits," *IEEE Signal Processing Letters*, vol. 9, no. 8, pp. 262–265, August 2002.

[23] M. Goodwin and M. Vetterli, "Matching pursuit and atomic signal models based on recursive filter banks," *IEEE Trans. on Signal Processing*, vol. 47, no. 7, pp. 1890–1902, July 1999.

[24] L. Daudet, "Sparse and structured decompositions of signals with the molecular matching pursuit," *IEEE Trans. on Audio, Speech and Language Processing*, pp. 1808–1816, September 2006.

[25] A. Bultan, "A four-parameter atomic decomposition of chirplets," *IEEE Trans. on Signal Processing*, vol. 47, no. 3, pp. 731–745, March 1999.

[26] R. Gribonval, "Fast matching pursuit with a multiscale dictionary of Gaussian chirps," *IEEE Trans. on Signal Processing*, vol. 49, no. 5, pp. 994–1001, May 2001.

[27] R. Gribonval and E. Bacry, "Harmonic decomposition of audio signals with matching pursuit," *IEEE Trans. on Signal Processing*, vol. 51, no. 1, pp. 101–111, January 2003.

[28] K. Kashino and H. Murase, "A sound source identification system for ensemble music based on template adaptation and music stream extraction," *Speech Communication*, vol. 27, pp. 337–349, March 1999.

[29] E. Vincent, "Musical source separation using time-frequency source priors," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 14, no. 1, pp. 91–98, January 2006.

[30] J. Eggink and G. J. Brown, "A missing feature approach to instrument identification in polyphonic music," in *Proc. of IEEE Int. Conf. on Audio, Speech and Signal Processing (ICASSP)*, Hong Kong, April 2003, pp. 553–556.

[31] V. Temlyakov, "Weak greedy algorithms," *Advances in Computational Mathematics*, vol. 12, no. 2,3, pp. 213–227, 2000.

[32] S. Krstulovic and R. Gribonval, "Mptk: matching pursuit made tractable," *Proc. of IEEE Int. Conf. on Audio, Speech and Signal Processing (ICASSP)*, May 2006.

[33] P. Leveau and L. Daudet, "Multi-resolution partial tracking with modified matching pursuit," in *Proc. of European Signal Processing Conference (EUSIPCO)*, 2006.

[34] G. Forney Jr, "The Viterbi algorithm," *Proc. of the IEEE*, vol. 61, no. 3, pp. 268–278, March 1973.

[35] M. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *J. Res. Nat. Bur. Stand*, vol. 49, no. 6, pp. 409–436, 1952.

[36] P. Jost, P. Vandergheynst, and P. Frossard, "Tree-based pursuit: Algorithm and properties," *IEEE Trans. on Signal Processing*, vol. 54, no. 12, pp. 4685–4697, December 2006.

[37] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Musical Instrument Sound Database," distributed online at http://staff.aist.go.jp/m.goto/RWC-MDB/.

[38] (auteur inconnu), "Iowa database." [Online]. Available: http://theremin.music.uiowa.edu/MIS.html

[39] Ircam, "Studio online database." [Online]. Available: http://forumnet.ircam.fr/402.html?L=1

[40] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: an Algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, November 2006.

[41] E. Chew, "Modeling Tonality: Applications to Music Cognition," *Proceedings of the 23rd Annual Meeting of the Cognitive Science Society*, pp. 206–211, 2001.

[42] N. Hu, R. Dannenberg, and A. Lewis, "A Probabilistic Model of Melodic Similarity," *Proc. of Int. Conf. on Computer Music (ICMC)*, 2002.

[43] J. Brown, "Computer identification of musical instruments using pattern recognition with cepstral coefficients as features," *Journal of the Acoustical Society of America*, vol. 105, p. 1933, 1999.

[44] K. Martin, "Sound-Source Recognition: A Theory and Computational Model," Ph.D. dissertation, Massachusetts Institute of Technology, 1999.

[45] A. Eronen and A. Klapuri, "Musical instrument recognition using cepstral cofficients and temporal features," 2000.

[46] S. Essid, "Classification automatique des signaux audio-fréquences: reconnaissance des instruments de musique (in french)," Ph.D. dissertation, Université Pierre et Marie Curie, 2005.

[47] T. Kinoshita, S. Sakai, and H. Tanaka, "Musical sound source identification based on frequency component adaptation," in *Proc. IJCAI Worshop on Computational Auditory Scene Analysis*, 1999, pp. 18–24.

[48] E. Vincent and X. Rodet, "Instrument identification in solo and ensemble music using independent subspace analysis," in *Proc. of Int. Conf. on Music Information Retrieval (ISMIR)*, 2004.

[49] E. Vincent, "Modèles d'instruments pour la séparation de sources et la transcription d'enregistrements musicaux," Ph.D. dissertation, University Pierre et Marie Curie, 2004.

[50] P. Jinachitra, "Polyphonic instrument identification using independent subspace analysis," in *Proc. of Int. Conf. on Multimedia and Expo (ICME)*, 2004.